

Appl. No. 10/773,673  
Amdt. dated August 21, 2006  
Reply to Office Action of April 19, 2006

### Remarks

The present amendment responds to the Official Action dated April 19, 2006. A petition for a one month extension of time to respond and authorization to charge Deposit Account No. 50-1058 the large entity extension fee of \$120 accompany this amendment. The Official Action objected to claims 9-11, 13,15, 17, 18, 20-23, 25, and 26 because of informalities. The Official Action rejected claim 17 under 35 U.S.C. §112 as being indefinite. The Official Action rejected claims 9 and 18-26 under 35 U.S.C. §102(e) based on Sheaffer U.S. Patent No. 6,957,321 (Sheaffer). The Official Action rejected claims 10, 11, 13, and 15 under 35 U.S.C. §103(a) based on Sheaffer in view of Moller et al. U.S. Patent No. 6,826,522 (Moller). The Official Action rejected claims 12, 14, and 16 under 35 U.S.C. §103(a) based on Sheaffer/Moller in view of Tremblay U.S. Patent No. 6,341,348 (Tremblay). The Official Action rejected claim 17 under 35 U.S.C. §103(a) based on Sheaffer in view of Oinaga U.S. Patent No. 4,665,479 (Oinaga). These grounds of rejection are addressed below.

Claims 1-8 were withdrawn by oral election. Claims 9, 11-23, 25, and 26 have been amended to be more clear and distinct. Claims 9-26 are presently pending.

### Objections

The Official Action objected to claims 9-11, 13,15, 17, 18, 20-23, 25, and 26 because of informalities concerning the meaning of "instruction slot". The independent claims 9, 17, 18, 20, and 21 have been amended to clearly define the meaning of "instruction slots".

Appl. No. 10/773,673  
Amdt. dated August 21, 2006  
Reply to Office Action of April 19, 2006

The Official Action also objected to the use of the acronyms as not being clear. The following acronyms used in the claims are consistent with their use in the specification:

Very long instruction word = VLIW page 4, line 10

VLIW memory = VIM page 4, line 21, page 7, line 22, and page 9, line 11

VLIW instruction memory basket = VIM Basket = VIMB page 4, line 22 and page 11, line 13

Indirect VLIW = iVLIW page 5, line 5

Load iVLIW = load indirect VLIW = LIV page 15, line 22

Because the use of these acronyms are consistent with their use in the specification and since the claims must be read in light of the specification, the acronyms have not been changed.

#### Section 112 Rejection

The Official Action rejected claim 17 under 35 U.S.C. §112 as being indefinite as to the meaning of the word "basket". The VLIW instruction memory basket (VIMB) is a special type of memory which stores VLIWs that may be fetched for execution as required by a program. In addition, a VLIW in the VIMB has "at least one instruction slot that is an expanded instruction slot" "wherein the width of the at least one expanded instruction slot is greater than the width of instructions required in program storage", as claimed in amended claim 17. Claim 17 has been amended to clarify the meaning and use of instruction slots and the VIMB and the relationship between the instruction slots and the VIMB.

Appl. No. 10/773,673  
Amdt. dated August 21, 2006  
Reply to Office Action of April 19, 2006

### The Art Rejections

Sheaffer, Moller, Tremblay, and Oinaga do not support the Official Action's reading of them and the rejections based thereupon should be reconsidered and withdrawn. Further, the Applicant does not acquiesce in the analysis of Sheaffer, Moller, Tremblay, and Oinaga made by the Official Action and respectfully traverses the Official Action's analysis underlying its rejections.

The Official Action rejected claims 9 and 18-26 under 35 U.S.C. §102(e) based on Sheaffer. Sheaffer describes an instruction set extension using operand bearing no operation (NOP) instructions as a way of extending the format of an individual instruction of an existing instruction set architecture. An instruction of an existing format in a program instruction stream is extended after a decode operation detects an associated operand bearing NOP instruction and thereby forms an expanded instruction. Both an existing instruction and its associated operand bearing NOP instruction must be decoded together before the benefits of the operand bearing NOP instructions can be obtained. This is explicitly shown in Sheaffer in Fig. 2B where step 225 states "Receive first instruction and operand specifying NOP instruction", the next step 235 states "Associate NOP operand(s) with first instruction and/or other instruction(s)", and the next step 245 states "Perform operation specified by first instruction using operands specified by NOP instruction". Sheaffer, Figs. 2A and 2B, col. 5, lines 55-65, col. 6, lines 14-15.

In contrast to Sheaffer, the present invention does not require a decoder, such as Sheaffer's decoder 205, to translate from one instruction set to another instruction set. As recited in amended claim 9, VLIWs, made up of instruction slots, are stored "at addressable locations in

Appl. No. 10/773,673  
Amdt. dated August 21, 2006  
Reply to Office Action of April 19, 2006

a VLIW memory from which VLIWs may be fetched for execution". In the VLIW memory of claim 9, a VLIW has "At least one expanded width instruction slot having a width that is greater than the standard width instructions slots". In order to store a VLIW, the VLIW memory is "configured for loading said at least one expanded width instruction slot with an expanded width instruction" as claimed in amended claim 9. Sheaffer does not teach or make obvious these and other various aspects of amended claim 9.

With respect to claim 18, it is further noted that the fetching of a VLIW and the loading of a VLIW are two different operations. As recited in claim 18, the fetching operation involves the use of "an instruction memory holding a plurality of instructions of a first bit width" "having at least one execute VLIW instruction" and "a very long instruction memory having slots for storing instructions of a second bit width". The "very long instruction memory holds VLIWs at addressable locations" that have been previously loaded and the loaded VLIWs reside in the very long instruction memory until they are fetched for execution. Furthermore, a VLIW "may be fetched as a result of executing the at least one execute VLIW instruction". A fetched VLIW already has "instruction slots of a second bit width wherein the second bit width is different from the first bit width". Sheaffer does not teach or make obvious a VLIW memory having VLIWs with different width instructions or the fetching of a VLIW with different width instructions from a VLIW memory.

Claims 20 and 21 have been amended in a similar fashion to claims 9 and 18 to clarify and distinctly claim various aspects of the invention, and therefore, are novel over and are not rendered obvious by Sheaffer at least for the reasons cited above.

Appl. No. 10/773,673  
Amdt. dated August 21, 2006  
Reply to Office Action of April 19, 2006

Since dependent claims 10-16, 19 and 22-26 depend from and contain all the limitations of the amended claims 9, 18 and 21, respectively, claims 10-16, 19 and 22-26 distinguish from the references in the same manner as claims 9, 18 and 21, respectively, placing claims 9-16, 18-19 and 20-26 in order for allowance.

The Official Action rejected claim 17 under 35 U.S.C. §103(a) based on Sheaffer in view of Oinaga. As addressed above, Sheaffer does not teach or make obvious a VLIW memory having VLIWs with an expanded width instruction or different width instructions, the fetching of a VLIW with an expanded or different width instruction from a VLIW memory, or other aspects of the invention as presently claimed. Oinaga describes a system that makes use of indirect address load/store instructions for the loading and storing of data. Indirect address load/store instructions are standard data accessing instructions which are used to generate indirect addresses to access data. Oinaga uses indirect address load/store instructions for the loading and storing of data between a main storage unit and vector registers. Oinaga, Fig. 2A and 2B, and col. 2, lines 15-29. In contrast to Oinaga, the present invention describes a load indirect VLIW (LIV) instruction used for loading instructions into a VLIW memory. Oinaga's data are not instructions and Oinaga's vector registers are not a VLIW memory. Oinaga does not teach or make obvious a load indirect VLIW (LIV) instruction and its use as presently claimed in amended claim 17. Oinaga does not cure the deficiencies of Sheaffer.

The Examiner also cites Oinaga as describing a "load mask bit field". Oinaga's mask registers 22 are used to "determine whether or not each vector operation should be carried out" Oinaga, col. 3, lines 61-63. Amended claim 17 claims that "an instruction bit organizer for

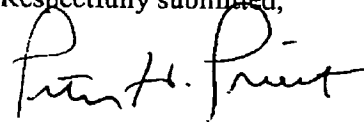
Appl. No. 10/773,673  
Amdt. dated August 21, 2006  
Reply to Office Action of April 19, 2006

receiving instructions as data and based on the load mask bit field organizing the bits from the data encoded instructions into proper format for loading into the specified instruction slots in the VIMB". The use of the "load indirect VLIW (LIV) instruction comprising a load mask bit field" to specify "which instruction slots are to be loaded in a VLIW having at least one instruction slot that is an expanded instruction slot, the VLIW accessible to be loaded at an addressable location in the VIMB" as claimed in amended claim 17 is distinctly different in concept, implementation, and function than masking off vector operations of Oinaga. Oinaga does not teach or make obvious various aspects of "a load mask bit field" as claimed in amended claim 17. Oinaga does not cure the above addressed deficiencies of Sheaffer.

#### Conclusion

All of the presently pending claims, as amended, appearing to define over the applied references, withdrawal of the present rejection and prompt allowance are requested.

Respectfully submitted,



Peter H. Priest  
Reg. No. 30,210  
Priest & Goldstein, PLLC  
5015 Southpark Drive, Suite 230  
Durham, NC 27713-7736  
(919) 806-1600